CS230

Semantic Segmentation of Extreme Climate Events

Romain Lacombe
rlacombe@stanford.edu
05432771

Hannah Grossman
hlg@stanford.edu
06487478

Lucas Hendren
hendren@stanford.edu
06493063

David Lüdeke
dludeke@stanford.edu
06173156

Abstract

Climate action failure and extreme weather events are the two most severe global risks today. To advance automated detection of extreme weather events, we have applied significant modifications to a novel light-weight context guided convolutional neural network, CGNet, and trained it for semantic segmentation of tropical cyclones and atmospheric rivers in climate data. In this project, primary interest was given to tropical cyclones, the most destructive extreme weather events, for which previous models showed poor performance. We investigated data feature engineering and augmentation, channel combinations, learning rate modifications, alternative loss functions, and architectural changes. We specifically chose to focus on recall and sensitivity metrics, in contrast to previous approaches focusing on IoU (intersection over union), to penalize under-counting and optimize for identification of tropical cyclones. Overall, we found success in improving these metrics through the use of weighted loss functions, and identified directions for future research. We hope to contribute to improved automated extreme weather events detection models, which are of crucial importance for better attribution, prediction and mitigation of the impacts of climate change.

1 Introduction

Climate action failure and extreme weather are the two most severe global risks today according to the World Economic Forum [1]. Studies of extreme weather and climate change rely on heuristics or expert judgment to identify weather events in climate simulations, which lead to discrepancies in their predicted frequency and less accurate attribution estimates [2]. Automated detection of extreme weather events in observational and simulated data could greatly accelerate research on the impacts of climate change. Since 2020, deep learning techniques for semantic segmentation have shown promise for that purpose [2]. However, they so far have relied on complex and heavy architectures with huge numbers of parameters. A key area of research is the development of lighter-weight architectures for semantic segmentation of tropical cyclones (TC) and atmospheric rivers (AR) [3].

In this project, **we apply the light-weight context guided convolutional neural network, CGNet, to semantic segmentation for the identification of tropical cyclones in climate data**. Input to our algorithm is hand-labeled climate simulation netCDF4 data with channels that contain key atmospheric variables such as wind speed, moisture content and atmospheric pressure for different times, latitudes and longitudes. The output is segmentation masks where each pixel takes a value corresponding to the background, TC, or AR classes.

Specific challenges include the very small dataset size, inherent class imbalance of relatively infrequent extreme events, unavoidable bias due to human-labeling, and limited capacity of the light-weight network. We report experiments with different hyperparameters (loss function, learning rate), architecture (upsampling), data augmentation, and feature engineering. **We find that weighted loss functions aimed at compensating class imbalance provide the most significant improvement on extreme weather events recall.**

CS230: Deep Learning, Fall 2022, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

2 Related work

Initial inspiration came from *ClimateNet: an expert-labeled open dataset and deep learning architecture for enabling high-precision analyses of extreme weather*, by Prabhat et al. [2], 2020, which implements DeepLabV3+ architecture. This ∼50M parameter model trained on human expert-labeled data achieved an IoU (equation 3) score of 0.24 for TCs, our primary class of interest. The study is the first to demonstrate that DL models trained on expert-labeled climate data can effectively perform semantic segmentation on extreme weather patterns. However, the large architecture is costly in terms of memory, training/inference time, and associated energy use.

Additional work was introduced in *Spatio-temporal segmentation and tracking of weather patterns with light-weight Neural Networks* by Kapp-Schwoerer et al. [3]., 2020. This group attempts to perform the same segmentation task on the same dataset as Prabhat et al., but utilizing the much lighter-weight architecture of CGNet (Figure 2). They improved on Prabhat et al.'s performance with a IoU score for of 0.34 for TCs, and a relatively low recall of 0.57 for TCs. This model and these metrics are the baseline performance for our project.

To more deeply understand the CGNet architecture used by Kapp-Schwoerer et al., we referenced the original paper that introduced the CGNet architecture *A light-weight context guided network for semantic segmentation*, by Wu et al. [4]. In searching for solutions to our class imbalance, we referred to various loss functions reviewed in Jadon 2020 [5]: *survey of loss functions for semantic segmentation*, to guide our loss landscape experiments. Lastly, we rely on *Deep Learning for the Earth Sciences* [6] for general background on applying deep learning techniques to Earth sciences and to explore training attribution models on weather events.

3 Dataset and Features

3.1 ClimateNet Dataset

To train our neural network, we used labeled climate data from *ClimateNet* [7], an open, community-sourced, human expert-labeled dataset which maps the outputs of Community Atmospheric Model (CAM5.1) climate simulation runs for 459 time steps from 1996 to 2013. Each example is a `netCDF` file containing an array $(1152, 768)$ for one time step, with each pixel mapping to a (latitude, longitude) point with 16 channels for key atmospheric variables, and one class label. The dataset is split in a **training set** of 398 (map, labels) pairs spanning years 1996 to 2010 in the CAM5.1 climate simulation, and a **test set** of 61 (map, labels) pairs spanning 2011 to 2013. For learning rate decay, we created a validation set of 56 (map, labels) pairs, which we set aside from the training set spanning 2008-2010 in order to keep the test set consistent with our baseline.

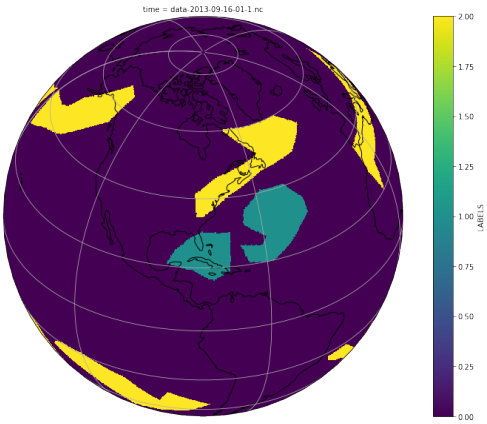

Figure 1: Segmentation labels (AR: yellow; TC: green).

We used the `xarray` library [8] to analyze and visualize the dataset. A description of the 16 channels (Table 2) and visuals of two channels with the human-labeled segmentation map for one data sample (Figure 5). The base model comes with a built in normalization step that we used in all subsequent model experiments.

The baseline implementation developed by Kapp-Schwoerer et al. utilizes the following four channels [3]:

1. **TMQ**: total vertically integrated precipitable water
2. **U850**: zonal (east-west) winds at the 850 mbar pressure surface
3. **V850**: meridional (north-south) wind at the 850 mbar pressure surface
4. **PSL**: atmospheric pressure at sea level.

The output of the model is a $(1152, 768)$ tensor of `softmax` probabilities for classes `'0'` for background, `'1'` for TC, or `'2'` for AR. Labels for the supervised learning of this task are segmentation maps, as illustrated in Figure 1. Importantly, these were hand-drawn by climate scientists as part of a community labeling exercise described in Prabhat et al., 2021 [2]. Figure 4 illustrates how labels were generated as a consensus between experts.

3.2 Data Engineering

From the existing 16 channels, we engineered new features *wind velocity* and *wind vorticity*, which we hypothesized would allow the model to more directly learn to identify TCs, which are characterized by high wind speeds and rotation.

Wind speed is the L_2 norm of zonal and meridional components of the wind vector field. Wind vorticity is the curl of the wind vector field around the earth radius axis, a measure of the local rotation [9]. These engineered features were computed at the 850hPa level to produce new channels WS850 (equation 1) and VRT850 (equation 2), and at the lowest level to produce WSBOT (equation 1) and VRTBOT (equation 2).

3.3 Data Augmentation

In an effort to reduce overfitting in the relatively small train set, we explored data augmentation techniques. The most promising involved transforming the image based on longitude. Although we explored random shifting and flipping, we discovered that random shifts to the longitudinal axis immediately decreased our metrics. We believe this is due to the importance of geography (relative positioning of continents and oceans) to regional climatic circulation and weather patterns. Rather than providing additional data, augmentation failed to reflect accurate geographical representations and was therefore a detriment to learning.

4 Methods

4.1 Baseline Implementation and Performance

We established a baseline by running an implementation [3] of the CGNet architecture. The network was trained for 15 epochs over the ClimateNet training set with a Jaccard loss (equation 5) based on the intersection over union (IoU) metric for the three classes background, AR, and TC.

In addition to IoU, we are optimizing for recall as a key confusion metric. It is especially important to minimize false negatives for identification of infrequent events. We find that the baseline prediction for TCs performs a IoU score of 0.34 and recall of 0.57 on the test set (as seen in Table 1). A higher performance on the train set (IoU score of 0.38 for TCs) indicates the model may also have variance.

A fundamental challenge for climate event identification is the inherent imbalance of the data, since, by definition, the extreme events we aim to detect are very rare. We conclude from this analysis that **our baseline implementation exhibits very high bias, some variance, and relatively low recall**.

4.2 CGNet Architecture

For our baseline, we chose to work with the light-weight CGNet architecture implemented by Kapp-Schwoere et al. [3]. CGNet follows the principle of "deep and thin" and is designed specifically for high-accuracy semantic segmentation while maintaining a small memory footprint. This is advantageous for reducing training time, memory footprint and model complexity.

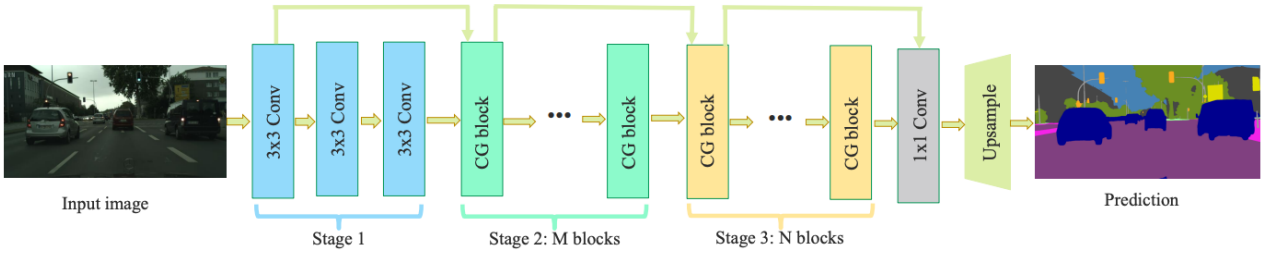

Figure 2: CGNet artchitecture using the CG blocks (figure from Wu et al. 2021 [4])

The basic unit of CGNet is a Context Guided (CG) block, which concatenates the output of normal and dilated convolutional layers to integrate local and surrounding context respectively. It heavily uses 1x1 convolutions and then average pooling to further refine the representation using a global context. The CG block reduces the number of parameters and memory footprint by employing channel-wise convolutions, thus removing computational cost across channels [4].

In order to facilitate our new methods, we tried including an additional CNN + BatchNorm + ReLU layer to the model to produce a deeper network that could potentially learn higher level and more complex features. We also experimented with doubling the final upsampling layer to further increase the size of the output predictions. Both of these attempts were unsuccessful at significantly increasing our metrics.

4.3 Loss Functions for Imbalanced Classes

The foremost challenge we faced is the extreme data imbalance inherent to rare weather events. Prahbat et al., 2021 [2] report 94% of pixels in the ClimateNet data belonging to the background class. We found that TCs represent only 0.462% of pixels of the entire dataset (and ARs only 5.674 %). This means that a naive model assigning *every pixel* to the background class would reach 94% accuracy, while obviously failing at its task. To address this class imbalance, we set out to experiment with modifying the loss landscape to better account for under-represented classes and improve performance on rare events such as TC and AR pixels. To that end, we leaned on the review of loss functions in Jadon, 2020 [5]. We selected, implemented, and used the following metrics and loss functions for training.

4.3.1 Metrics

As our problem statement was to identify rare events in climate, we explored performance metrics that would better represent the model's capacity to solve that task. Specifically, we value identifying events accurately more than identifying the exact boundaries of the segmentation hand-labeled by experts. We also aimed at penalizing missing extreme events more than over-predicting the geographical extent of events. We implemented the following success metrics:

- **Intersection over union:** our baseline model was trained to optimize for the IoU metric (equation 3), as usual for many computer vision problems.
- **Sørensen–Dice similarity:** or Dice coefficient (equation 4) is a measure of the similarity between class predictions and ground-truth that is widely used for image comparison.
- **Recall** or **Sensitivity:** we devised our training strategy to optimize for this metric as a proxy for the ability to detect most true positives of the TC class.

Loss functions To optimize for these metrics, we aimed to explore loss functions designed to over-weight rare classes. We explored and implemented the following:

- **Jaccard loss:** used by our baseline mode. Computes a derivable prediction of segmentation map IoU from the `softmax` probabilities output of the classifier (equation 5)
- **Cross-entropy (CE) loss:** classicially used in multiple classification problems, helps overweight under-represented class. We used the `pyTorch` CE (equation 6) and weighted CE (equation 7) losses.
- **Focal Tversky loss:** highly tunable loss function which overweighs false positive and negatives as well hard examples in the data, by introducing a power law parameter γ (equation 8).
- **Weighted Jaccard loss:** to normalize the relative weights of each class in the IoU estimate, we finally experimented with a custom loss function inspired by the Jaccard loss (equation 9).

5 Experiments/Results/Discussion

Exploring with learning rates greater (0.1, 0.01) and less (0.0001) than the default learning rate (0.001) negatively affected IoU and Dice scores. In order to deal with issues of over-fitting, we implemented a new learning rate scheduler for the Adam optimizer with decay and early termination (implemented in model #2). This proved successful in reducing the overfitting observed in the baseline implementation.

After the results of our experiments and comparing them for both precision-recall and specificity-sensitivity we found that both our weighted cross entropy and weighted Jaccard loss experiments, with feature engineering and the learning rate scheduler performed better than the baseline, and our tests with learning rate scheduling, feature engineering, and cross entropy loss performed as well or worse then the baseline.

We measured IoU, Dice, precision, recall/sensitivity and specificity both for TC and AR events, but we were specifically focused foremost on recall, specificity and TC events. TC because they are the more dangerous of the two, and for recall and specificity because we determined it was better to focus on catching all positives versus other metrics given the severity of a positive event.

We report the precision-recall graph as well as the sensitivity-specificity ROC curve in Figure 3. Raw data for the confusion metrics of the key models we trained are reported in Figure 1. We didn't include data augmentation in this analysis due to the significant decrease in performance.

We encountered and identified several issues and pitfalls, the first among which was the limited and imbalanced data which made it difficult to improve on task performance. We found some success however with our experiments on loss

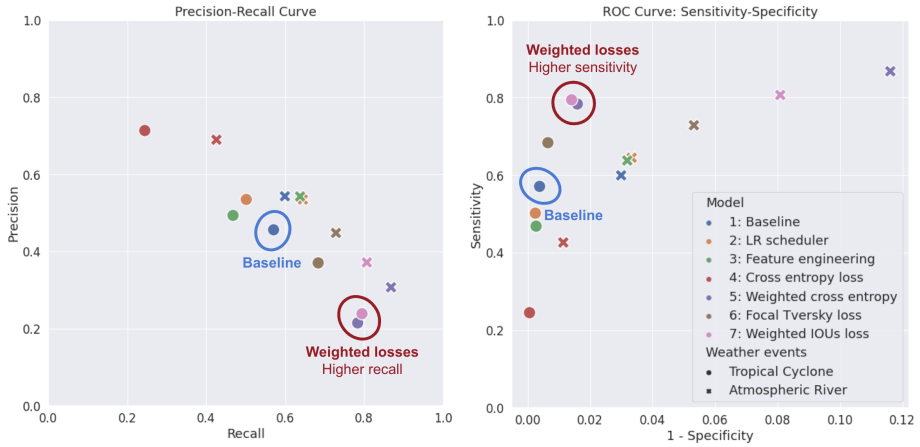

Figure 3: Precision vs Recall (left) and ROC Curve (Sensitivity vs 1-Specificity, right)

landscapes and functions. Secondly, this was an intentionally light model, which created limitations on complexity and model capacity.

Lastly, given the climate focus of this neural network and the goal of keeping its resources usage light weight, we tracked and evaluated our carbon footprint during our experimentations. Based on emissions factors from Lacoste et al. [10], 2019, and an estimated 40 hours of usage of a Nvidia A100 GPU with 40GB of RAM, we estimated our emissions for model training at around 6.24 kg CO_{2e}.

6 Conclusion/Future Work

In conclusion, this was a very challenging problem. Because advances in lightweight segmentation are so new (CGNet was released in 2020), we have found no applications of these novel architectures to climate data so far beyond the baseline we have reported. The small, biased, and imbalanced dataset made it difficult to produce a successful model. Our analysis indicates that IoU is a poor performance metric for identifying severe and rare climate events. Precision-recall have proved to be better indicators since recall prioritizes how many successful predictions were made for a class given its prevalence. Our results have shown that throwing more computing power at this problem doesn't improve the model or result in better performance. Instead, we demonstrated that modifications to the model and loss function for the task at hand can yield significant improvements. We found relative success by increasing the performance recall with new learning methods, features and architecture. That being said, with more time, we identified two key areas we would like to investigate further.

6.1 Future Work: More and Better Data

A critical issue is the excessively limited, imbalanced and biased dataset. While our experimentation with augmentation did not lead to improved results, likely due to geographical issues, more and better data could help improve learning. Utilizing much larger observational datasets matched with expert-labels or semi-supervised training could potentially increase the amount of data available, and improve performance.

6.2 Future Work: Transfer Learning

Lastly, as visible in Figure 4, individual labels seem to be highly subjective. We suspect the subjective human-expert labeling strategy may lead to **high unavoidable bias**, which may account for the limited baseline performance. To try and reduce Bayes error on this task, we would like to explore ways to create better ground-truth segmentation maps. We have identified a historical database of TCs and extents which may help for that purpose. If this experiment were to prove tractable, we may be able to use pre-trained baseline weights for transfer learning and generalization to observational weather data rather than simulations.

Contributions

Romain Lacombe contributed to dataset identification, exploration and analysis, literature search, baseline evaluation, features engineering, training on GPU, model training and performance analysis, weighted loss functions, carbon footprint evaluation, and project organization. **Hannah Grossman** contributed to environment & AWS setup, loss function updates, experimentation lead, architecture exploration, up-sampling investigation, project organization, setup for ablation study, data augmentation, and final video presentation. **Lucas Hendren** contributed to environment, Colab, & AWS setup, data Visualization and analytics, data augmentation, initial baseline setup, loss function updates, instance and debugging support. **David Lüdeke** contributed to experimentation execution, progressive model experimentation design, data visualization, precision-recall space exploration, model prediction evaluation, and final video slides. All authors contributed equally to this report.

Data, Code, and Models

We provide an [online repository] with:

- Our implementation of the CGNet model, building on Kapp-Schwoerer et al. [3].
- Notebooks for download, exploration, and visualization of the dataset, generation of engineered features, and flexible model training.
- Six models and a baseline ready to be loaded for inference, complete with configuration files, trained weights, and training and validation evaluation metrics history.

References

[1] World Economic Forum. *Global Risks Report*. 2022. URL `https://www.weforum.org/reports/global-risks-report-2022/`.

[2] Prabhat, K. Kashinath, M. Mudigonda, S. Kim, L. Kapp-Schwoerer, A. Graubner, E. Karaismailoglu, L. von Kleist, T. Kurth, A. Greiner, A. Mahesh, K. Yang, C. Lewis, J. Chen, A. Lou, S. Chandran, B. Toms, W. Chapman, K. Dagon, C. A. Shields, T. O'Brien, M. Wehner, and W. Collins. Climatenet: an expert-labeled open dataset and deep learning architecture for enabling high-precision analyses of extreme weather, 2021. URL `https://gmd.copernicus.org/articles/14/107/2021/`.

[3] Lukas Kapp-Schwoerer, Andre Graubner, Sol Kim, and Karthik Kashinath. Spatio-temporal segmentation and tracking of weather patterns with light-weight neural networks. *AI for Earth Sciences Workshop at NeurIPS 2020.* URL `https://ai4earthscience.github.io/neurips-2020-workshop/papers/ai4earth_neurips_2020_55.pdf`.

[4] Tianyi Wu, Sheng Tang, Rui Zhang, Juan Cao, and Yongdong Zhang. CGNet: A light-weight context guided network for semantic segmentation. *IEEE Transactions on Image Processing*, 30:1169–1179, 2021. doi: 10.1109/TIP.2020.3042065. URL `https://github.com/wutianyiRosun/CGNet`.

[5] Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, oct 2020. doi: 10.1109/cibcb48159.2020.9277638. URL `https://doi.org/10.1109%2Fcibcb48159.2020.9277638`.

[6] Mayur Mudigonda, Prabhat Ram, Karthik Kashinath, Evan Racah, Ankur Mahesh, Yunjie Liu, Christopher Beckham, Jim Biard, Thorsten Kurth, Sookyung Kim, Samira Kahou, Tegan Maharaj, Burlen Loring, Christopher Pal, Travis O'Brien, Kenneth E. Kunkel, Michael F. Wehner, and William D. Collins. *Deep Learning for the Earth Sciences*. John Wiley Sons, Ltd, 2021. doi: https://doi.org/10.1002/9781119646181. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119646181`.

[7] Lukas Kapp-Schwoerer, Andre Graubner, Sol Kim, and Karthik Kashinath. ClimateNet, a Python library for deep learning-based climate science. 2020. URL `https://github.com/andregraubner/ClimateNet`.

[8] S. Hoyer and J. Hamman. Xarray: N-D labeled arrays and datasets in Python. *J. Open Res. Software*, 2017. URL `https://xarray.pydata.org/en/v0.9.2/index.html`.

[9] Isla Simpson. Circulation and vorticity (class lecture, advanced atmospheric dynamics, university of toronto), 2010. URL `https://www2.cgd.ucar.edu/staff/islas/teaching/3_Circulation_Vorticity_PV.pdf`.

[10] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning, 2019. URL `https://arxiv.org/abs/1910.09700`.

Appendix

Results

Models:		1: Baseline	2: Learning rate decay	3: Feature engineering	4. Cross entropy	5. Weighted cross entropy	6. Focal Tversky	7. Weighted Jaccard
TC	**IoU**	0.33955127	0.3491679	0.31608774	0.22278776	0.20251324	0.31599551	0.22453519
	Dice	0.5069627	0.51760482	0.48034448	0.36439318	0.33681665	0.48023798	0.36672721
	Precision	0.45598923	0.53463995	0.49327914	0.71342764	0.21450748	0.37011321	0.23838463
	Recall	0.57076677	0.50162175	0.46807083	0.24468465	0.78363352	0.6836551	0.79444291
	Specificity	0.99623709	0.99758723	0.99734295	0.99945687	0.98414286	0.9935705	0.98597405
AR	**IoU**	0.39832633	0.41285328	0.4146876	0.35750965	0.29317686	0.38387118	0.34108058
	Dice	0.5697187	0.58442485	0.58626032	0.52671397	0.45342113	0.55477878	0.5086653
	Precision	0.54289729	0.5343576	0.54252858	0.68965182	0.30685734	0.44789329	0.37141691
	Recall	0.59932803	0.64484427	0.63766037	0.42605401	0.86800497	0.72866865	0.80679887
	Specificity	0.97010985	0.96671544	0.96815083	0.98864332	0.88386163	0.94679579	0.91912138

Table 1: Metrics: baseline and 6 selected models we trained.

ClimateNet Dataset

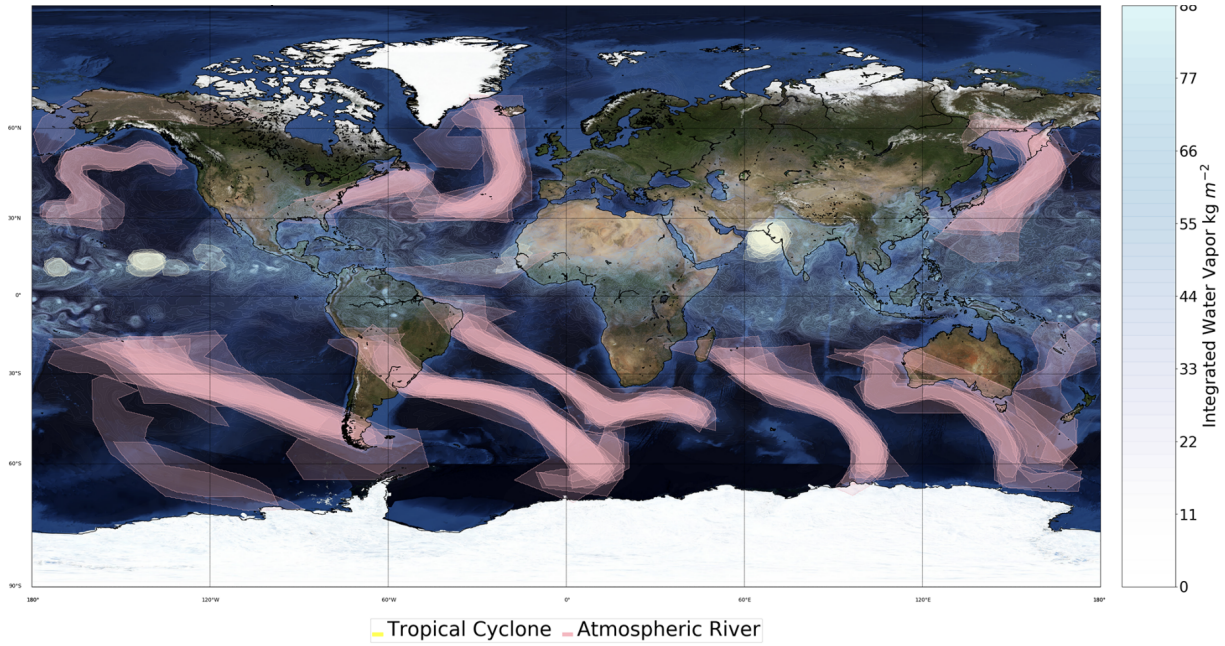

Figure 4: Example image from Prabhat et al. 2021 showing 15 different expert labelings (TC labels in white/yellow masks seen near the equator; AR labels in pink masks). The background "bluemarble" map included via Matplotlib's Basemap library is © NASA [2].

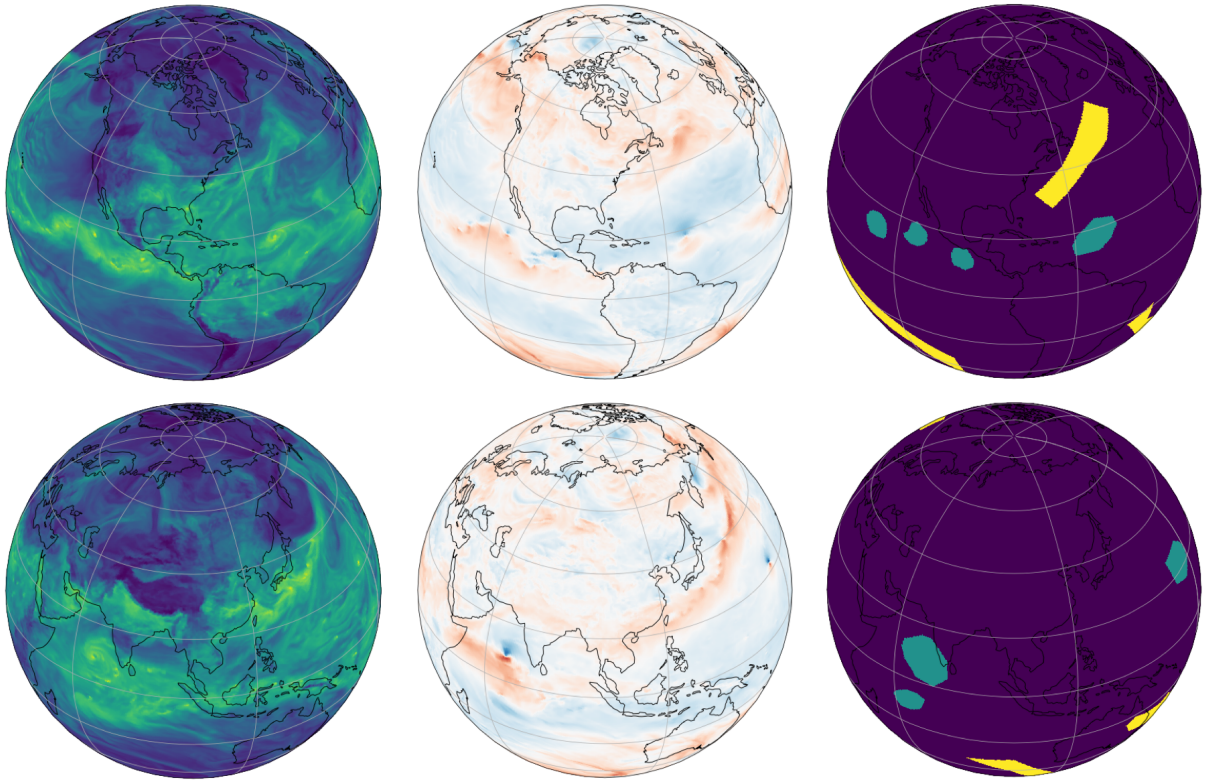

Figure 5: Channels TMQ, U850 and segmentation labels (AR: yellow; TC: green) for a single example in the ClimateNet training set, viewed from 35°N and 80°W (above) and 35°N and 100°E (below).

Channel	Description	Units
TMQ	Total (vertically integrated) precipitable water	kg/m^2
U850	Zonal wind at 850 mbar pressure surface	m/s
V850	Meridional wind at 850 mbar pressure surface	m/s
UBOT	Lowest level zonal wind	m/s
VBOT	Lowest model level meridional wind	m/s
QREFHT	Reference height humidity	kg/kg
PS	Surface pressure	Pa
PSL	Sea level pressure	Pa
T200	temperature at 200 mbar pressure surface	K
T500	temperature at 500 mbar pressure surface	K
PRECT	total (convective and large-scale) precipitation rate	m/s
TS	surface temperature (radiative)	K
TREFHT	reference height temperature	K
Z1000	geopotential Z at 1000 mbar pressure surface	m
Z200	geopotential Z at 200 mbar pressure surface	m
ZBOT	lowest modal level height	m
LABELS	0: Background, 1: Tropical Cyclone, 2: Athmospheric river	-

Table 2: ClimateNet dataset channels and labels [7].

Equations

Engineered Feature: Wind Velocity

Wind speed is the L_2 norm of the zonal and meridional components of the wind vector field:

$$w_s = \sqrt{u^2 + v^2} \tag{1}$$

Engineered Feature: Relative Wind Vorticity

Wind vorticity is the rotation of the wind vector field, where λ = longitude and ϕ = latitude.

$$\zeta = \frac{\partial u}{\partial \lambda} - \frac{1}{\cos \phi} \frac{\partial v \cos \phi}{\partial \phi} \tag{2}$$

Metrics & Loss Functions

Here we present the cost functions for a single sample. Formalism: $y = y_{ij}$ is a one-hot encoded ground truth tensor for the 3 classes over latitude and longitude (i, j), and $\hat{y} = \hat{y}_{ij}$ is the 3-classes probabilities tensor computed as the `softmax` of the logits predicted by the network. Parameters are w_C, the tensor of weights used to overponderate imbalanced classes, and β and γ, scalars which allow for the tuning of relative weights of false positives and false negatives and of hard examples in the focal Tversky loss. All operations here are element-wise.

$$\text{INTERSECTION OVER UNION} = \frac{|X \cap Y|}{|X \cup Y|} \tag{3}$$

$$\text{DICE LOSS}(y, \hat{y}) = 1 - \frac{2y\hat{y} + 1}{y + \hat{y} + 1} \tag{4}$$

$$\text{JACCARD LOSS} = 1 - \frac{|X \cap Y|}{|X \cup Y|} = 1 - \frac{\hat{y}y}{(\hat{y} + y) - \hat{y}y} \tag{5}$$

$$\text{CROSSENTROPY LOSS}(y, \hat{y}) = -y \log(\hat{y}) \tag{6}$$

$$\text{WEIGHTED CROSSENTROPY LOSS} = -w_C y \log(\hat{y}) \tag{7}$$

$$\text{FOCAL TVERSKY LOSS}(y, \hat{y}) = \left(1 - \frac{y\hat{y}}{\beta(1-y)\hat{y} + (1-\beta)y(1-\hat{y})}\right)^{\gamma} \tag{8}$$

$$\text{WEIGHTED JACCARD LOSS} = 1 - w_C \frac{\hat{y}y}{(\hat{y} + y) - \hat{y}y} \tag{9}$$

Segmentation Results

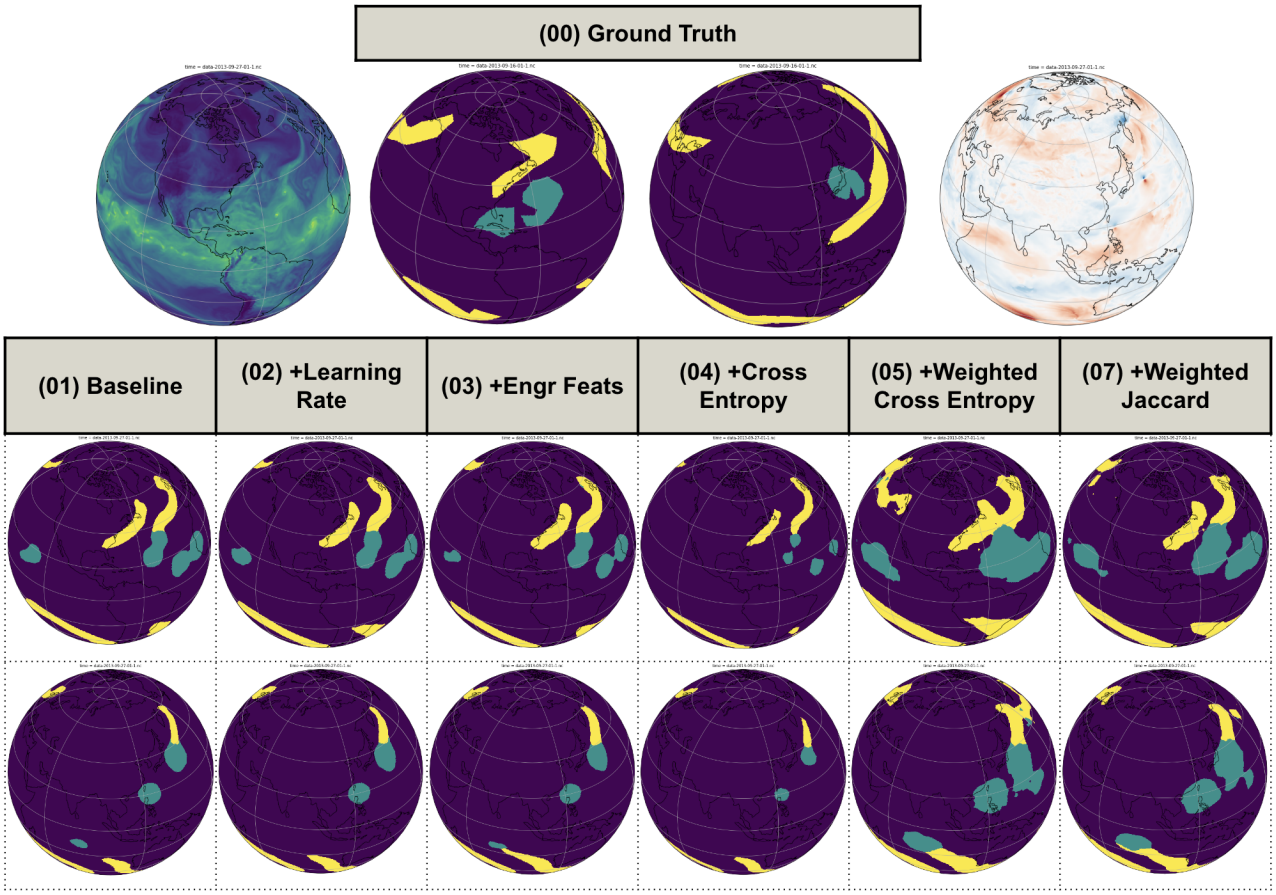

Figure 6: Final results visualizations, displaying the ground truth labels at the top, then the prediction masks produced by the baseline and 5 of our models. The first row of globes shows North America and the second row shows Eurasia.